

TCP/IP 소켓을 이용한 실시간 감시 시스템 설계 및 구현

Design and Implementation of Real-Time Surveillance System using TCP/IP Socket

이종서*, 김유두*, 문일영*

Jong-Seo Lee*, Yu-Doo Kim* and Il-Young Moon*

요 약

근래에 원격지 감시와 제어하는 기술이 발달하고 있다. 또한 홈 네트워크 서비스의 보편화와 초고속 통신망의 폭넓은 확대에 인하여 디지털 비디오 기술을 이용한 감시체계가 점차 늘어나고 있다. 웹에서 제공하던 감시 체계를 사용자가 좀 더 편리하게 사용할 수 있도록 기존의 웹을 기반으로 하던 디지털 감시 체계에 개인 단말기를 이용하여 사용자가 이동 중에도 인터넷이 되는 곳에서 이 모듈을 제어할 수 있도록 구현하였다.

Abstract

Recently, remote observation and control technology has grown gradually. According to use home network system and expand super-highway information network, digital video technology observation system gradually has increased. In order that the user conveniently use better than to use web browser observation system, it uses existing digital observation system, personal digital assistant and control module in the place of internet.

Key words : TCP/IP Socket, Real-Time Surveillance, Home Observation

I. 서 론

최근 사회의 각 분야에 있어서 원격 제어 및 원격 감시 기술은 통신과 컴퓨터 기술의 급속한 발전으로 우리의 생활을 변화시키고 있다[1].

또한 최근 유비쿼터스 시대로 도래하면서 홈 네트워크 서비스가 크게 각광 받고 있으며, 가정내의 모든 정보 가전 기기가 유무선으로 연결되어 누구나 기기, 시간, 장소에 구애받지 않고 다양한 디지털 홈 서비스를 제공받을 수 있는 미래 지향적인 가정환경을 구축하고 있다.

이런 사회적 변화에 따라 기존의 아날로그방식의

감시카메라 영상녹화 방법은 점차 사라져 가고, 초고속 통신망의 등장으로 고속의 화상처리 서비스가 가능하게 되어 다양한 서비스를 제공하는 환경이 구축되어 웹 기반 원격 영상 감시 및 제어시스템의 실현이 되고 있다. 또한 디지털 비디오 기술을 이용한 원격 상황 감시 시스템으로 상황실에서 원격지의 상황을 직접 관리할 수 있게 되었고, 제어 시스템 장애 발생시 원격지 영상 및 음향 신호의 도움으로 신속하게 상황을 판단하거나 각 장치들의 운용상태 확인, 비정상 사건 상황에 따라 전송하는 요구가 확산되고 있다.

본 논문에서는 TCP/IP 소켓을 이용하여 홈 내 상

* 한국기술교육대학교 인터넷미디어공학부(School of Internet Media Eng., Korea university of Technology and Education)

· 제1저자 (First Author) : 이종서

· 접수일자 : 2007년 12월 20일

황에 대해 실시간으로 감시를 통해서 개인단말기를 이용하여 언제 어디서나 사용할 수 있는 편리를 도모하고 실제 적용하도록 한다.

제 2장에서는 이번 시스템에서 쓰인 영상 압축 기술과 TCP/IP 소켓 기술을 살펴보고, 제 3장에서는 시스템의 설계를 살펴본다. 제 4장에서는 시스템 구현을 살펴보고, 제 5장에서는 결론과 향후 연구 과제를 통해 제안하는 시스템이 적용 되어질 방향을 제시하도록 한다.

II. 구현기술

2-1 영상압축기술

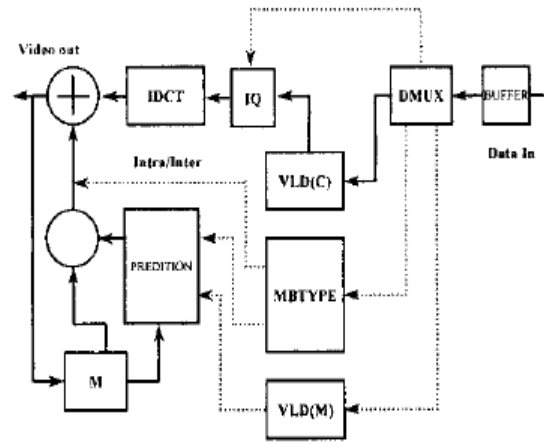
본 모듈에서는 H.263과 JPEG압축기술을 이용하여 영상압축을 하였다. 2장에서는 압축기술과 TCP/IP를 소개하도록 하겠다[2].

(1) H.263 압축기술 : H.263표준은 화상통신을 위해서 만들어진 것으로 ITU-T표준이다[3]. 저비트율 통신을 위해 설계 되었고 초기에는 64Kbits/s 보다 낮은 전송률을 표준으로 삼았으나 저비트율 뿐만 아니라 광대역 비트율에서도 사용할 수 있도록 하고 있다.

H.263 보다 낮은 데이터 전송 속도의 영상 회의나 텔레비전 전화 실현을 목적으로 하는 컬러영상신호의 디지털 부호화 기술을 규정한 ITU-T에서 권고한 H.261이 통신 회선으로서 종합정보통신망(ISDN)을 전제로 하고 있다면, H.263에서는 일반 전화망을 상정하고 있다. 따라서 데이터 전송속도를 V.34 모뎀의 28.8kbps 이하로 했다. 영상 부호화 기술은 H.261에서 사용하고 있는 것을 그대로 시행하고 있으며, 다시 개량을 거듭하여 H.261과 동일한 정도의 영상 품질을 실현했다. H.263의 기술은 MPEG 4 표준 규격의 기본으로 사용된다.

본 모듈에서는 소스 코덱을 통해 압축이 이루어진다. 압축된 스트림은 클라이언트로부터의 요구가 있을 경우 일정한 크기의 패킷단위로 잘라서 TCP/IP를 통해 보내진다.

그림 1은 소스디코더에 대한 구성도를 보여준다.



VLD(C) : Variable Length Decoder for transform coefficients.

VLD(M) : Variable Length Decoder for motion vectors.

M : Decoded frame store.

그림 1. H.263 소스 디코더

Fig. 1. H.263 Source Decoder

(2)JPEG 압축기술 : JPEG는 정지 영상 압축 전용 프로세서를 장착한 이 시스템은 Baseline 순차 프로세서로 Discrete Cosine Transform(DCT)을 기초로 하고 Variabe length 부호화(Huffman 부호화)를 한다. 이 프로세서들은 압축 정도가 4:1 - 100:1 까지 압축이 제공되며 복원 시 영상의 시각적 특성이 높은 충실도를 제공한다[4].

JPEG 표준인 ISO/IEC 10918은 오늘날 가장 많이 사용되고 있는 사진 압축 포맷이다. 이 표준은 상당히 높은 압축율의 고품질 사진이나 매우 높은 압축 비율의 적정한 품질의 사진 중에서 선택할 수 있는 유연성을 가지고 있다. 카메라나 화면과 같은 시스템은 이 기술의 단순함으로 인해 저렴하게 제작될 수 있다.

2-2 TCP/IP 소켓

소켓(Socket)이란 TCP/IP 프로토콜을 이용하여 프로그래밍 언어에 대한제한 없이 데이터를 주고받기 위해 만들어진 공통 API(Application Programming Interface)이다[5]. 이를 활용하면 C, JAVA등 언어에 상관없이 각 언어에 맞게 제공되는 API를 활용하여 TCP/IP 프로토콜의 네트워크를 통해 데이터를 주고 받을 수 있다[6].

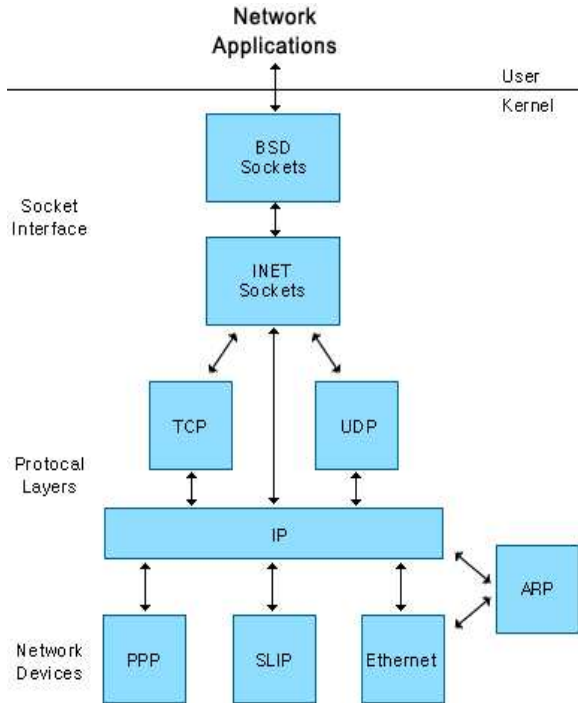


그림 2. 네트워크 계층
Fig. 2. Network Layers

그림 2는 네트워크 계층도 이다. 여기서 TCP/IP는 신뢰할 수 있는 일대일 프로토콜로서, 데이터를 주고 받기 위해 IP 프로토콜을 사용한다. IP 패킷에 헤더가 붙어 있는 것처럼, TCP 패킷에도 헤더가 붙어 있다. TCP는 연결 중심적인 프로토콜로 두 네트워크 응용 프로그램은 그 사이에 많은 서버 네트워크, 게이트웨이 및 라우터가 있더라도 단일한 가상의 접속을 통해 연결된다. TCP는 두 응용 프로그램 간의 데이터를 신뢰할 수 있는 방식으로 전달하며 데이터의 손실이나 중복이 없다는 것을 보장한다.

III. 시스템 설계

본 논문에서 구현한 시스템의 전반적인 구조와 각각의 서버, PC 클라이언트, PDA 클라이언트 간의 상호작용하는 모습을 살펴보도록 한다.

3-1 시스템 구조

전반적인 시스템의 구성은 MFC(Microsoft Foundation Class)로 구현하였다.

그림 3은 시스템의 전체적인 구조를 보여주고 있다. 서버, PC 클라이언트와 PDA 클라이언트 총 3가지로 구성되어 있으며 각각의 모듈들은 TCP/IP 소켓으로 연결되어 서로간의 통신을 한다.



그림 3. Home Observation의 시스템 구조
Fig. 3. Structure of Home Observation

3-2 서버/PC클라이언트 구조

서버와 PC 클라이언트는 구현상의 큰 차이점이 없도록 설계되었다. 모듈을 실행할 때 서버와 PC 클라이언트를 선택하는 과정에서 두 모듈의 차이가 생기게 된다.

그림 4에서 보면 이 두 모듈은 TCP/IP 소켓으로 연결되어 있다. 클라이언트는 서버가 있는 원격지의 IP 주소로 서버에 접속을 하여 서버에 설치되어있는 CAM 을 이용하여 원격지의 디지털 영상정보를 수신하게 된다. 또한 클라이언트는 서버의 CAM을 제어할 수 있도록 통신 서로간의 메시지를 설정하여 원격지의 CAM을 제어한다.

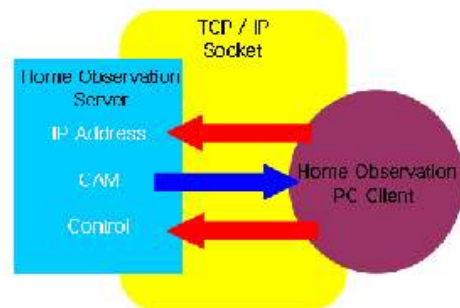


그림 4. 서버, PC클라이언트 세부시스템 구조
Fig. 4. System Structure of Server and PC Client

3-3 서버/PDA 클라이언트 구조

PDA는 클라이언트는 원격지의 서버의 고유 IP를 가지고 서버로 접속한 다음 서버의 CAM을 제어할 수 있다.

원격지의 영상을 받아 오지는 않지만 원격지의 CAM을 제어 할 수 있도록 설계되었다. 그림 5는 서버, PDA 클라이언트의 시스템 구조를 보여준다.

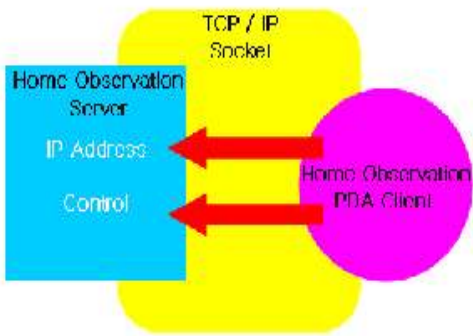


그림 5. 서버,PDA 클라이언트 구조
Fig. 5. Structure of Server and PDA Client

IV. 시스템 구현

실시간 감시 시스템에서는 기본적으로 CAM이 설치되어 있어야 한다. 한편 PC 클라이언트는 CAM을 가지고 있지 않아도 된다. 서버는 CAM을 인식하고 제어할 수 있어야 하며 서버와 PC클라이언트는 받아 온 영상을 원하는 압축 방식에 따라서 Decoding, Encoding을 하여야 한다.

표 1은 CAM을 제어하는 부분이다.

표 1. CAM 제어부분
Table 1. Control Part of CAM

```

BOOL CCapture::Initialize()
{
    // 비디오 캡처 윈도우를 생성
    m_hCapWnd = capCreateCaptureWindow( "Capture",
        WS_POPUP, 0, 0, 1, 1, 0, 0 );
    if( m_hCapWnd == NULL ){
        return FALSE;
    }
}
    
```

```

capSetUserData( m_hCapWnd, this );
// 캡처 이벤트 콜백 함수 지정
capSetCallbackOnVideoStream(m_hCapWnd,
OnCaptureVideo );
// 캡처 윈도우와 장치 드라이버 연결 (장치 ID : 0)
if( !capDriverConnect( m_hCapWnd, 0 ) ){
    m_hCapWnd = NULL;
    return FALSE;
}
// 캡처 정보 설정
if( SetCaptureParam() == FALSE ){
// 캡처 장치 드라이버 해제
    capDriverDisconnect( m_hCapWnd );
    return FALSE;
}
return TRUE;
}
    
```

표 2는 구현기술에서 설명한 H.263의 Encoding 부분을 나타낸다.

표 2. H.263 압축방식
Table 2. H.263 Compression form

```

void CNetExDlg::H263Encoding( BYTE *data, int size )
{
    m_nQFactor=GetDlgItemInt( IDC_EDIT_QFACTOR );
    Bits bits;
    // YUV 형식으로 영상을 변환함

    ConvertRGB2YUV(IMAGE_WIDTH,IMAGE_HEIGHT,data,yuv);
    count=0;
    // H263 압축 설정
    cparams.format =m_nCompressFormat;
    cparams.inter = CPARAM_INTRA;

    cparams.Q_intra = m_nQFactor;
    cparams.data = yuv; // 영상 데이터 포맷 설정 YUV

    CompressFrame( &cparams, &bits ); // H263 압축

    // 영상 데이터 전송
    SendImageData( cdata, count, 3 );
}
    
```

표 3은 JPEG압축방식의 Encoding을 나타낸 것이다.

표 3. JPEG 압축방식

Table 3. JPEG Compression form

```

void CNetExDlg::JpegEncoding( BYTE *data, int size )
{
    m_nQFactor=GetDlgItemInt( IDC_EDIT_QFACTOR );
    CString strFile = _T("temp.jpg");
    int BMPWidth = (IMAGE_WIDTH*3+3)/4*4;
    int bWidth = IMAGE_WIDTH;
    int bHeight = IMAGE_HEIGHT;
    int i = 0, j = 0;
    for( i = 0 ; i < bHeight ; i++ )
    {
        for( j = 0 ; j < bWidth * 3 ; j++ )
        {
            int idx1 = ( ( bHeight - 1 ) - i)*BMPWidth + j;
            int idx2 = i * BMPWidth + j;
            m_TmpBuf[ idx1 ] = data[ idx2 ];
        }
    }
    // JPEG 압축
    CJpeg JpegEnc;
    JpegEnc.EncodeJpg((LPSTR)(LPCTSTR)strFile,
    IMAGE_WIDTH,IMAGE_HEIGHT,m_TmpBuf,m_nQFactor,
    m_ImageBuf, m_ImageSize );
    // 영상 데이터 전송
    SendImageData( m_ImageBuf, m_ImageSize, 2 );
}
    
```

압축된 영상은 다시 PC 클라이언트에서 Decoding을 걸쳐 화면에 다시 나타나게 된다.

PDA 클라이언트에서는 서버의 상태를 제어하여 모듈을 실행시키거나 종료를 할 수 있다.

그림 6은 Home Observation 시스템의 서버 실행한 화면이다.



그림 6. 서버 실행모습
Fig. 6. Playing Server

그림 7은 서버에 IP주소로 접속을 한 후 영상을 받은 PC 클라이언트의 실행화면이다.



그림 7. PC클라이언트의 실행모습
Fig. 7. Playing PC Client

그림 8은 서버모듈을 제어하는 PDA 클라이언트의 모습이다.



그림 8. PDA 클라이언트의 실행모습
Fig. 8. Playing PDA Client

위의 그림과 같이 3가지의 모듈의 실행 모습을 살펴 보았다.

V. 결 론

본 논문에서는 TCP/IP 소켓을 이용하여 원격지의 상태를 감시하는 시스템을 구현해 보였다. 원격지에 대한 실시간 시스템은 위험한 지역에 있어서의 제어 및 감시에 도움을 줄 뿐만 아니라, 산업에 적용하여 산업재해를 미연에 방지 할 수 있는 효과도 클 것으로 기대한다.

앞으로의 향후과제로서는 이동성이 뛰어난 PDA 클라이언트와 서버가 영상 전송을 통한 감시체계가 좀 더 구현되어야 할 것이며, PDA 뿐만 아니라 핸드폰기기로 이 기능을 확장 한다면 CAM과 핸드폰만 있다면 이 시스템을 쉽게 구축하며 실생활에서 적용할 수 있을 것이다.

참 고 문 헌

[1] 성경상 외, “영상 스트리밍 기술을 이용한 홈 서버 스텝 감시 및 제어 시스템”, *한국인터넷정보학회*, 제5권 2호, pp.269~272, 2004.11

[2] 압축기술, <http://www.axis.co.kr/html/camerall/Docu/compression.pdf>.
 [3] 이성수 외, “H.263을 기반으로 하는 인터넷 동영상 서비스 시스템 구현”, *대한전자공학회*, 제 21권 1호, pp.737~740, 1998.01
 [4] 진순중 외, “고압축 JPEG 영상을 위한 블록킹 현제거”, *한국통신학회*, 제 31권 제 9C호, pp.869~875, 2006.09
 [5] 조경민 외, “TCP/IP 소켓 프로그래밍”, 한빛미디어, 2005.07
 [6] MFC 프로그래밍, <http://yalli.new21.org>.

이 종 서 (李鍾瑞)



2003년 3월~현재 : 한국기술교육대학교 인터넷미디어공학부 재학
 관심분야 : 모바일 콘텐츠, 모바일 프로그래밍

김 유 두 (金裕斗)



2007년 2월 : 한국기술교육대학교 인터넷공학(공학사)
 2007년 3월~현재 : 한국기술교육대학교 대학원 정보미디어공학과 (석사과정)
 관심분야 : 무선 네트워크, 모바일 콘텐츠, 스마트카드

문 일 영 (文日永)



2000년 2월 : 한국항공대학교 항공통신정보공학과 (공학사)
 2002년 2월 : 한국항공대학교 대학원 항공통신정보공학과 (공학석사)
 2005년 2월 : 한국항공대학교 대학원 정보통신공학과 졸업(공학박사)

2004년~2005년: 한국정보문화진흥원 선임연구원

2005년 3월~현재 : 한국기술교육대학교 인터넷미디어공학부 조교수

관심분야 : 무선 인터넷 응용, 무선 인터넷, 모바일 IP